Object Oriented Programming, Exercise 3

Topics: Increase coding routine in Python, using Git, pseudocode, a bit of classes and objects

Make a Git commit at least after every coding task.

Code in Python3 and follow the style guide.

1. Explain the following terms:
    a. Abstraction (in programming)
    b. Accessor and mutator methods
    c. Public and private methods
    d. __str__ method (in Python)
2. Modify the Coin class (see Exercise2) so that in addition to **sideup** you have another data attribute called **currency**. Add a function generating the currency (Euro, Pound, Dollar, Ruble, Yen). Use a random generator to get the currency (=similar to tossing the coin). Add a function to print out the currency.
3. Add a method that can change the currency of the coin. Test that your coin still works.
4. Change the Coin's sideup attribute to private. Test that your coin still works. What happens, if you now try to change the attribute's value from the main function? Try it out...!
5. Create a class **Dice** and make an object of it. You shall be able to roll the dice, get the result (number between 1 – 6) and get its color. Add at least 1 extra feature. Design your program using pseudocode. Document your code properly (with good comments) and pay attention to the clarity of the output prints.
6. Create two Dice objects and roll them both. Sum the result and print to screen.
7. Design first using pseudocode, then code this: Create a Dice rolling game of three players (three Dice objects). On first round everybody rolls their dice, lowest number loses and is out of game. On second round the two remaining contestants roll a dice and higher number wins. Use proper output prints of the situation all the time. If on either round there is a tie between 2 or 3 dices, then the tied dices are rolled again.
8. Design first using pseudocode, then code this: Create a CellPhone Class. Write a program that will design a class that represents a cell phone. The data attributes are manufact (Manufacter), model (Model) and retailPrice (Retail price). The class will also have the following methods:
    a. __init__
    b. set Manufact
    c. set Model
    d. setRetailPrice
    e. getManufact
    f. getModel
    g. getRetailPrice

Test your CellPhone Class by writing a main function that creates the needed objects and prompts the user for the phone's manufacturer, model and retail price. Program Output with data that the user enter:

h. Enter the manufacturer: Apple
i. Enter the model number: 'iPhone7
j. Enter the retail price: 500
k. Here is the data that you provided :
l. Manufacturer: Apple
m. Model number: iPhone 7
n. Retail price: 500.0

9. Take a look at the CellPhone Class/Object: where are these concepts (or are they there) (take a screen capture and indicate a line)?
    a. Object?
    b. Encapsulation?
    c. Data attributes?
    d. Hidden attributes?
    e. Public methods?
    f. Private methods?
    g. Init-method?